

### C. Remarks

The presently pending claims stand rejected variously based on the newly cited reference Aman et al (US Patent 6,249,800) and the prior cited references Primak et al (US Publication 2002/0010783), Ballard (US Patent 6,078,960), Mangipudi et al (US Patent Publication 2004/0162901), and Kawata (US Publication 2002/0032777). Applicants respectfully assert that the cited art, as cited in combination, fails to teach or suggest the presently claimed invention.

The teachings of Primak, Kawata, Ballard and Mangipudi were summarized in the prior Responses.

#### Rejection of Claims 1 - 6 in view of Kawata and Primak:

The rejection of Claim 1 is maintained effectively based on a combination of three references: Kawata, a reference cited in the background section of Kawata (Hei 11-250020; hereafter the "Hei" reference), and Primak. These references, properly considered, do not teach or suggest the present invention as set forth in Claim 1. Specifically, the references fail to teach or suggest how they can be combined in any proper and meaningful way and, even if combined, fail to teach or suggest all required aspects of the method of Claim 1.

#### The Teachings of the Kawata Reference, for the Purposes Cited, Are Incompatible

First, the Hei reference described in ¶13 of Kawata, as previously summarized, requires the servers to initiate and periodically upload a server load status value to a third system, a state management server. This state management server is separate from both the clients and servers. The load status value, as generated by a server, is expressly defined as a measurement of the "number of IP packets per unit time" handled by that server. The Hei reference teaches that each server periodically "informs [the] state management server of its own load status." The state management server is a passive receiver with the various servers

being independently self-directed in posting load status values to the state management server.

Counting undifferentiated IP packets only as a function of time does not distinguish on the basis of client requests and, therefore, the load status as held by the state management server or as queried by clients cannot be not determined against any particular client request.

So, all that Hei teaches is that servers should periodically post packet count-based load values to a third server and clients should query that third server for the single lowest load value.

Kawata teaches a quite different and, in fact, incompatible system. At ¶37, Kawata expressly teaches that clients blindly direct all requests to a “load balancer” device separate from both the clients and servers:

[S]ervice requests sent from the clients 105 go through a load balancer 100, which distributes the service request from the clients 105 to the servers.

The Kawata clients simply do not actively participate in the target server selection process.

If, as suggested in the Action, Kawata were modified by Hei to have the clients request load values from the load balancer, either that operation would be meaningless or the load balancer would cease to function as intended by Kawata. In the first instance, since all requests would still go through the load balancer for distribution, there would be no need to inform the load balancer of the server with the lowest load value; the load balancer would already know. In the second instance, if the requests were not directed to the load balancer,

it would not function to “distribute[] the service request[s] from the clients 105 to the servers” as intended by Kawata.

Further, Kawata expressly teaches away from relying on the servers to determine and provide load status:

[S]ince the load balancing according to the present invention estimates the server load resulting from the service request packet, excessive communication load between servers and the load balancer and excessive server CPU load are prevented. (¶41; emphasis added.)

As discussed in the prior Responses, Kawata relies only on pre-computed load estimates to independently presume, entirely by the load balancer, the current loading of the different servers. These estimates are obtained from static data tables held internal to the load balancer. While the load balancer does update an aggregate load value for a server each time a request is forwarded to that server, absolutely no loading information is obtained through communication with that server. The estimate applied may have no current relation to the performance of the server.

Naturally, this aspect of Kawata completely contradicts the operating principle of Hei, which expressly requires the servers to initiate and periodically provide load status values to the state management server.

As the courts have repeatedly held, if a proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. In re Gordon, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984); In re Ratti, 270 F.2d 810, 123 USPQ 349 (CCPA 1959) (If the proposed modification or combination would change the principle of operation of the

prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious).

Ultimately, the teachings of the references are simply incompatible, at least in any manner relevant to the present invention as set forth in Claim 1.

It should be noted that the Primak reference is cited solely as teaching that a “selected server evaluates the client’s request” and “evaluates its capability to determine whether it is suitable to handle a client’s request.” (Action ¶14). To the extent presented in this Action regarding the teachings of Primak, Applicants agree. As previously discussed, and as implicitly acknowledged by the Action, Primak teaches or suggests nothing with respect to a client evaluating a client request in any relation to differentially selecting a server based on loading considerations. Primak provides no aid towards curing the mutual incompatibilities of the Hei and Kawata references.

#### The References Fail to Teach or Suggest All Required Elements of Claim 1

Again, independent Claim 1 requires:

A method of cooperatively load-balancing a cluster of server computer systems for servicing client requests issued from a plurality of client computer systems ...

- a) selecting, by a client computer system, a target server computer system ... to service a particular client request using available accumulated selection basis data ...
- b) evaluating, by said target server computer system, said particular client request to responsively provide instance selection basis data dynamically dependent on the configuration of said target server computer and said particular client request; and
- c) incorporating, by said client computer system, said instance selection basis data into said available accumulated selection basis data ... .  
(Emphasis added.)

Neither the Hei nor Kawata reference teaches or suggests that any client accumulate “selection basis data.” Contrary to the assertion made in the Action at ¶41, Claim 1

explicitly requires accumulation of "selection basis data" by the client computer system through the incorporation of "said instance selection basis data into said available accumulated selection basis data." The feature, as termed in the Action, of accumulating selection basis data to serve in the selection of a target server by a client computer system is a clear and explicit element of the claim.

Neither the Hei nor Kawata reference teaches or suggests that any selection of a target server proceed based on the client's own evaluation of the server loading with respect to a particular request. As claimed, the client accumulation of "selection basis data" is as a function of particular client requests. Per element (b), "instance selection basis data" is dynamically generated by the server with respect to a particular client request. Per element (c), as amended for clarification purposes only, the client computer system receives the "instance basis selection data," which the client computer system then incorporates as the "available accumulated selection basis data."

Neither the Hei nor Kawata reference teaches or suggests that any target server determine and report load information as "instance selection basis data" generated "dynamically dependent on the configuration of said target server computer and said particular client request." As required by Claim 1, this instance selection basis data is returned to a client computer to enable selection of a target server. As previously established, the clients of Hei retrieve nothing more than a single load status value for each of the servers, given that the load value is just a function of IP packets per unit time. The clients of Hei have no access to loading values in any way differentiated by client requests. As also previously established, the clients of Kawata simply do not actively participate in server selection.

Consequently, Claim 1 calls for a method whose steps include required operations not taught or suggested by the cited references. These differences between claim and

references represent significant, new operative capabilities. As explained in the present specification, the affirmative distribution of the “instance selection basis data” to the client computer systems enables an active interoperation – “cooperatively load-balancing” – between a plurality of client systems relative to a separate plurality of server systems without requiring direct communications between the clients or a third server to coordinate the load balancing. The clients can be geographically distributed without requirement of mutual communication and without a centralized load-balancing hub, yet realize an effective load-balancing of requests handled by the collection of target servers. Therefore, a system implemented according to Claim 1 operates fundamentally different from and is a substantial improvement over the cited load-balancing systems.

Given that the references thus do not teach or suggest the present invention as set forth in Claim 1, reconsideration of the rejection of Claim 1, including for the same reasons dependent Claims 2-6, is respectfully requested.

Rejection of Claims 7 - 19, 25 - 27, and 29 - 30 in view of Mangipudi and Aman:

In the present Action, the Aman reference is substituted for the Kawata reference. While teaching a technically different system, the Aman reference is substantively little different from the Kawata reference in terms of showing the present invention as set forth in Claim 7.

Summary Analysis of Aman:

Specifically, the Aman reference again utilizes a third server, a routing node 110E, to perform balancing of business operations handled by a system complex through the assignment of client work and session requests to specific individual application servers 115 within the server complexes 110A-D. Notably, Aman itself recognizes that, while participating as part of a larger system complex (sysplex) each of the application servers 115

operate and are properly considered as separate servers (col 10, ll 65-67; col 11, ll 44-67). The routing node 110E is also recognized as a fully separate server (col 12, ll 1-2).

The routing node 110E internally maintains certain system management tables dynamically updated to track, in short, the currently assigned work and session requests being processed by the different server complexes 110A-D (col 13, ll 31-47). Table entries are initialized by the routing node 110E as different application servers 115 and entire server complexes 110A-D register with the routing node 110E (col 13, ln 59 - col 14, 12). Subsequently, these complexes 110A-D report to the routing node 110E "the current utilization, in terms of service unit sums and percentages of total capacity, for that particular system" (col 14, ll 15-21). The service unit sums represent abstracted units of the work and session requests being processed over defined time periods. Further, these service unit sums and capacity percentages are apparently aggregated and reported for full complexes 110A-D, rather than individual application servers 115.

Consequently, as with Kawata, the balancing decision performed by the routing node is not informed by the actual loading of particular application servers and not with respect to any particular request. Rather, the balancing decision is made, at best, based on a loosely aggregated reflection of individual server loads – relying on load averages over, as a practical matter for high-performance computer systems, lengthy 60 second intervals further aggregated for multiple application servers, is a fundamentally loose relationship. In effect, these aggregated averages are, in terms of accuracy and timely relevance, functionally the same as the estimated values employed in Kawata; rather than being based on static precomputed values, the estimates of Aman are based on gross averages collected over broad time frames and generically from the complexes 110A-D.

As noted in the Action, certain values, incidentally denoted "weights" by Aman, are communicated by the routing node to the various clients to control the selection of target

servers by the clients. Specifically, in response to a particular work or session request issued by a client, the routing node generates and returns a fixed server list, with each listed server having an assigned fixed weight. The client then uses the list to direct correspondingly weighted shares of that work or session request to the different servers on the list (col 8, ll 33-39).

The routing node computes these weights based on the number of application servers eligible to process a given client work or session request and a complex function of the estimated capacity of the eligible application servers (col 7, ln 36 - col 8, ln 32). In essence, the routing node itself implements the complete load balancing algorithm based on the current estimated work loads being handled by the different application servers without actually consulting the individual application servers. The routing node exclusively determines the list of application servers identified to a client. The routing node exclusively determines the fractional client work load that is to be distributed to the different listed application servers. The client is provided with and has no knowledge of the specific loading of any application server or any other aspect of the burden imposed on an application server in performing any particular client request. Indeed, under Aman, a given client can potentially be directed to send a greater percentage portion of work to an application server with a higher actual loading and a lesser portion to an application server with a lower actual loading.

In any event, Aman clearly teaches that the clients themselves do not select application servers based on their own evaluation of application server loading. Rather, the Aman clients are only able to distribute fixed portions of the work or session request to the fixed set of application servers in exactly the manner specified by the router node. The Aman clients are incapable of discretion.

Again, Independent Claim 7 requires:

- A method of load-balancing a cluster of server computer systems in the cooperative providing of a network service to host computers operating mutually independent of one another ...
- a) selecting, independently by each of a plurality of host computers, server computers within a computer cluster ...
  - c) receiving, in regard to said respective service requests ... load and weight information from respective said server computers, wherein load and weight information is dynamically generated by respective said server computers; and
  - d) evaluating, by each of said plurality of host computers, respective load and weight information ... as a basis for a subsequent performance of said step of selecting. (Emphasis added.)

Claim 7 requires the provision of both "load and weight information" from the servers to a requesting "host computer," i.e., client, in response to a "respective service request." This "load and weight information" is required by Claim 7 to be "dynamically generated by respective said server computers." Claim 7 is further specific that the "load and weight information" be received by the host computers that originate the "respective service requests."

As before, no combination of Mangipudi and Aman teaches or suggests the retrieval of any other information, i.e., "weight information," from the application servers and certainly not "weight information ... dynamically generated ... by said server computers." Furthermore, there is no suggestion presented in Mangipudi or Aman that any other information relevant to load-balancing even exists on or can be retrieved from the servers.

To be sure, the "weight" values disclosed in Aman only define the percentage portions of a work or session request that are to be sent to corresponding application server. The "weight" information required by Claim 7 represents a value generated by the server, in addition to a load value, that is further communicated to the client. As required by the claim,

both “load and weight information” is used to independent select, by that client, a particular server to receive a particular service request.

Since Aman does not teach or suggest “weight information ... dynamically generated by ... said server computers” as required by the claim, Applicants respectfully assert that Claim 7 is not obvious in view of Mangipudi and Aman. Reconsideration of the rejection of Claim 7, as well as dependent Claims 8-12 for at least the same reasons presented in regard to Claim 7, is also requested.

Similar to Claim 7, independent Claim 13 requires:

- a) a plurality of server computers individually responsive to service requests ..., wherein said server computers are operative to initially respond to said service requests to provide load and weight values, wherein said load and weight values represent a current operating load and a policy-based priority level of a respective server computer relative to a particular service request; and
- b) a host computer system operative to autonomously issue said service requests [... and ...] to select a target server computer ... to receive an instance of said particular service request based on said load and weight values. (Emphasis added.)

For the reasons advanced above with respect to Claim 7, Applicants respectfully assert that Claim 13 and dependent Claims 14 - 19 are not obvious in view of Mangipudi and Aman. The claim explicitly requires “load and weight values” to be provided by a server computer to a host computer for use by the client computer in “selecting a target server computer [to receive a] particular service request.”

While Aman does disclose use of a “weight,” it is comparable to the presently claimed “load and weight values” in name only. The Aman weights are fixed values computed by the router node, at least from the perspective of the client, used by the client only to ration

corresponding percentage portions of a work or session request to a fixed list of application servers.

Accordingly, Applicants respectfully request reconsideration of the rejection of Claims 13 - 19.

Independent Claim 25 requires:

- c) receiving from said particular server computer system with respect to said particular client request instance selection qualification information discretely determined by said particular server computer system dynamically with respect to said particular client request, wherein said instance selection qualification information including a load value reflective of the current performance capability of said particular server computer system and a weight value reflective of the anticipated performance capability of said particular server computer system with respect to said particular client request, wherein said instance selection qualification information is incorporated into said accumulated selection qualification information. (Emphasis added.)

Similar to Claim 1, Claim 25 requires a load-balancing performed cooperatively by the clients and servers. This cooperative relation is archived by the servers evaluating "particular" client requests and returning to the clients "instance selection qualification information" that is specific to the "particular client request." This "instance selection qualification information" is accumulated and used by the client as the basis for selection of a particular server computer system for receipt of a particular client request.

Similar to Claim 7, Claim 25 further requires the server computer systems to return both load and weight information to the clients: said instance selection qualification information including a load value reflective of the current performance capability of said particular server computer system and a weight value reflective of the anticipated performance capability of said particular server computer system with respect to said particular client request.

Consequently, Mangipudi and Aman fail to teach or suggest the present invention as set forth in Claim 25. Reconsideration of the rejection of Claim 25, including for the same reasons dependent Claims 26-27, is respectfully requested.

Dependent Claim 29, as amended, specifies that:

said weight value part of said instance selection qualification information includes a relative prioritization of said particular client request with respect to said particular server computer system.

As established above, neither Mangipudi nor Aman teaches or suggests the dynamic determination of "a weight value reflective of the anticipated performance capability of said particular server computer system with respect to said particular client request," as set forth in independent Claim 25. Claim 29 qualifies the weight value as including a "relative prioritization," considering the "particular client request" in specific correspondence with the "particular server computer system."

The load-balancing performed by Mangipudi is dependent on only load values from the servers. The Mangipudi servers do not provide any information dynamically determined by a server that represents any "relative prioritization" of a "particular client request" with respect to a "particular server computer system."

Consequently, Claim 29 and Claim 30, as dependent therefrom, are not obvious in view of the combined teachings of Mangipudi and Aman. Accordingly, Applicants respectfully request reconsideration of the rejection of Claims 29 and 30.

Rejection of Claims 20, 22-24, 31, and 33-36 in view of Mangipudi, Ballard, and Aman:

Independent Claim 20 requires:

a) ... a server computer of said first plurality provides a response, including dynamically determined load and weight information, in

acknowledgment of a predetermined service request issued to said server computer system ...

- b) ... wherein said client computer system is reactive to said response ... and wherein said client computer system is responsive to said load and weight information of said response in subsequently autonomously selecting said first and second server computer systems.

Independent Claim 31 requires:

- c) second processing said particular client request ... by said particular target server system to dynamically generate instance selection information including a load value for said particular target server system and reflective of a combination of said particular client request and said particular target server system and a relative weighting value reflective of the combination of said particular client request and said particular target server system; and
- d) incorporating said instance selection information into said accumulated selection information for subsequent use in said step of selecting, wherein said step of selecting matches said particular client request, including said attribute data, against corresponding data of said accumulated selection information to choose said particular target server system based on a best corresponding combination of relative weighting value and load value.

As established in the prior Responses, neither the Mangipudi nor Ballard reference teaches or suggests the server production of both “load and weight information” or, indeed, any server “dynamically determined” information “in acknowledgment of a predetermined service request” as required by Claim 20.

As further established above, the Aman reference only teaches the use of a “weight” to specify the percentage of work that a client is to send to each of a fixed list of application servers. Both the list of eligible servers and the weight for each listed server is exclusively determined by the routing node, operating as an external load balancer. Even then, the computation of these server weights by the routing node is not based on information dynamically generated by and received from the application servers themselves. The Aman

routing node operates based only on the collected averaged and aggregated performance of the server complexes.

Nothing in the combination of the references serves to suggest, let alone motivate, a person of ordinary skill the art to consider having any server contribute dynamic instance selection information specific to particular service requests for use in a load-balancing algorithm performed among and with the active participation of a first plurality of server computers and a second plurality of client computers, where a client computer itself operates to "autonomously select a first server computer ... to which to issue said predetermined service request [based on] said load and weight information."

Particularly in regard to Claim 31, the load-balancer use of the "dynamically generate[d] instance selection information" is expressly required. When considering the "accumulated selection information," the "particular client request" is matched to the selection information to find a most preferred server for the client request based on "a best corresponding combination of relative weighting value and load value." Such a consideration of the server generated instance selection information, including both load and weighting information, is nowhere taught or suggested by the cited references, either alone or in combination.

Accordingly, Applicants respectfully request reconsideration of the rejection of Claims 20 and 31 as obvious in view of Mangipudi, Ballard and Aman. Reconsideration of the rejection of Claims 22-24 and 33-36, for at least the same reasons presented in regard to Claims 20 and 31, is also requested.

Conclusion:

In view of the above Amendments and Remarks, Applicants respectfully assert that Claims 1 – 20, 22 – 27, 29– 31, and 33 – 36 are now properly in condition for allowance. The Examiner is respectfully requested to take action consistent therewith and pass this application on to issuance. The Examiner is respectfully requested to contact the Applicants' Attorney, at the telephone number provided below, in regard to any matter that the Examiner may identify that might be resolved through a teleconference with the Examiner.

Respectfully submitted,

Date: 7/27/2007

By: Gerald B. Rosenberg  
Gerald B. Rosenberg  
Reg. No. 30,320

NEWTECHLAW  
285 Hamilton Avenue, Suite 520  
Palo Alto, California 94301  
Telephone: 650.325.2100